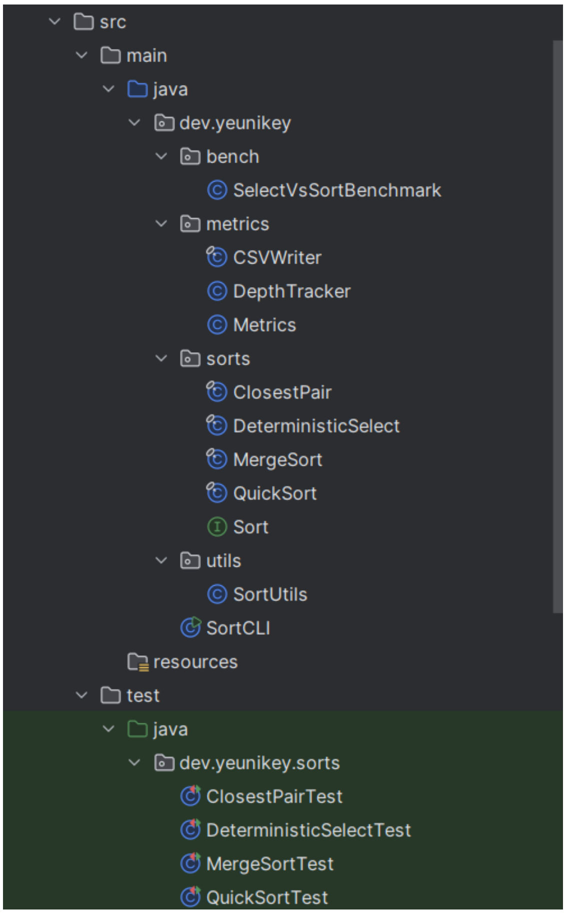
Assignment 1 – Divide and Conquer Algorithms

Yerassyl Unerbek

Project Structure



```
src
  main
    java
      dev.yeunikey
        bench
          SelectVsSortBenchmark
        metrics
          CSVWriter
          DepthTracker
          Metrics
        sorts
          ClosestPair
          DeterministicSelect
          MergeSort
          QuickSort
          Sort
        utils
          SortUtils
        SortCLI
    resources
  test
    java
      dev.yeunikey.sorts
        ClosestPairTest
        DeterministicSelectTest
        MergeSortTest
        QuickSortTest
```

Implemented Algorithms

1. **MergeSort** (Case 2 of Master Theorem)
    a. Linear merge with reusable buffer
    b. Small-n cut-off using Insertion Sort
2. **QuickSort** (robust version)
    a. Randomized pivot selection
    b. Recurse on smaller partition first, iterate on larger
3. **Deterministic Select (Median-of-Medians)**

    a. Group elements by 5, use median-of-medians as pivot
    b. In-place partitioning
    c. Recurse only on the required side (prefer smaller side)

4. **Closest Pair of Points (2D)**
    a. Sort by x-coordinate, recursive split
    b. “Strip” check by y-order (scan 7–8 neighbors)

Metrics

For each algorithm, the following data is collected:

- **Execution time**
- **Recursion depth**
- **Number of comparisons**

Results are exported to CSV for further analysis.

Analysis

Each algorithm includes:

- Recurrence relation and solution (Θ-notation)
- Analysis method (Master Theorem / Akra–Bazzi intuition)
- Comparison between **theory** and **measured results**

Testing

- Sorting correctness on random and adversarial arrays
- Verify recursion depth is bounded
- Selection verified using `Arrays.sort(a)[k]`
- Closest Pair validated against O(n^2) brute-force (for small n ≤ 2000)

Build & Test

```
# Build the project
mvn clean package

# Run all tests
```

```
mvn test
```

How to Run

```
# Run the main application (after build)
java -jar assignment-1.0-SNAPSHOT.jar \
    --size 10000 \
    --trials 3 \
    --output results.csv \
    --algo quicksort \
    --rnd 42
```

Parameters:

- `--size <N>`: Number of elements to process (array size for sorting algorithms or number of points for ClosestPair).
- `--trials <N>`: Number of times the algorithm runs for averaging metrics.
- `--output <filename>`: CSV file to save results.
- `--algo <name|all>`: Algorithm to run. Supported values:
    - `quicksort`
    - `mergesort`
    - `deterministicselect`
    - `closestpair`
- `--rnd <seed>`: Random number generator seed for reproducibility.

Benchmarking

```
mvn exec:java -Dexec.mainClass=org.openjdk.jmh.Main "-
Dexec.args=dev.yeunikey.bench.SelectVsSortBenchmark -bm avgt -wi 5 -
i 3 -rf csv -rff results.csv"
```

Documentation

Full documentation and source code are available on GitHub:
https://github.com/yeunikey/Assignment1DAA